Templates WP2
Guidelines for authors

V. Acary, F. Dubois

Siconos Work package 2 - Numerical Methods/ Software Development
Vincent.Acary@inrialpes.fr,dubois@lmgc.univ-montp2.fr

Version 1.1,
April 14, 2006

Purpose of this document This document provides guidelines for authors who want to propose a template for WP2.

1 Goal of a collection of templates

In requiring a collection of templates for WP2, several goals are expected :

- to provide a list and a description of various particular and well-known non smooth dynamical system (NSDS) which belong to the general family of NSDS,

- to outline the majors steps in the simulation of a physical or abstract NSDS

- to show how a given physical problem can be formalized into a well-identified class of abstract NSDS,

- to show the numerical strategies usually used to simulate a particular type of system (steps in the numerical procedure, operators which are needed, etc ....)

- to provide a collection of basic examples to validate logically the architecture of the platform

- to list what kind of results are needed by users.

We want to detect the genericity between the various cases of NSDS to define the classes (abstraction, factorization of attributes and methods).

The goal of collecting these templates is neither to gather an exhaustive list of problem where non smooth modeling is relevant, nor to treat in details a particular problem. The collection of templates is different from a collection of benchmarks where numerical efficiency (cpu time, accuracy, . . .) is tested. The collection of templates will be used to design the platform and to logically validate the architecture.

2 Guidelines for authors

2.1 What is a template ?

A template describes the different steps required to model numerically a given NSDS.
It should describe :

1. **[optional]** *The description of a physical problem.*
   This part is optional due to the fact the template may be a full abstract problem.

2. **[required]** *The definition of the general abstract class of NSDS used in this template*
   This part contains a general description of the kind of problem that will be treated in the template. If this general abstract class is already used in an other template, it is sufficient to refer to it. At the end, all these general abstract classes will be collected in a reference document in the software user manual. This aspect is very important to identify the different classes supported by the platform.

3. **[required]** *Formalization into a general abstract class*
   This part must contain a formalization of the problem in a compatible form with the general abstract class. It should be also interesting to detail the various steps to transform the initial "natural"form into this abstract form.

4. **[required]** *Description of the numerical strategy*
   This part will detail the different ingredients of the solving strategy (time integration, numerical methods ...). If an xxxLab file, which allows to solve the given problem, exists it should be added.

5. **[optional]** *Description of the results*
   A short description of the relevant results which have to be given by the algorithm

6. **[optional]** *Comparison with analytical or experimental results*
   If some analytical or experimental results are available, they should be provided in order to transform this template in a benchmark.

A complete template must contain all the required parts (2. 3. 4.) to be relevant. One can propose some "draft" of template only containing the part 3. (and 2. if possible) in order to encourage subsequent works to define numerical methods to complete part 4.

Remark 1 *For instance, if we want to simulate a walking robot, we can provide a Lagrangian model with constraints where all variables are, in a certain sense, anonymous with respect to the physical problem. Alternatively, we can give directly a physical description of the robot (rigid bodies with mass and geometry, constraints, initial conditions etc) and expect that the software performs the formalization. This remark explains why the first item may be optional.*
We make the distinction between these two types of descriptions because in the case of a direct description of the physical problem, the software must know how to transform a physical description of the problem into an abstract general problem. The way to transform a physical description is not defined by default in the platform and must be supplied by the user (functions, existing modeling software, offline computation, . . .).

In the Sections 2.3–2.7 , we provide guidelines for the drafting of the different part defined above

2.2   Format of templates

A complete template is composed of two parts :

1. **a description document**: a brief document (maximum 10 pages) which describes the template,

2. **a xxxlab simulation file**: a Matlab or a Scilab file which provides an example of simulation.

2.3   The description of a physical problem.

In the case of a description of a physical problem, the author should provide all the ingredients which are necessary to solve the problem.

2.4   The definition of the general abstract class of NSDS

In this part, the author describes briefly the general class of NSDS which corresponds to the particular problem, he wants to simulate, for instance :

- Lagrangian dynamical system with constraints

- Linear complementarity system

- Piece-wise linear system, . . .

For the description of NSDS, it seems to be more convenient to separate, if possible, three majors features:

1. The smooth dynamical part with the boundary conditions (IVP or BVP), for instance :

   - Linear time invariant system
   - Non linear system

- Lagrangian system, ...

2. The set of output/input variables which are involved in the non smooth law and theirs relations with the state vector of the dynamical system

3. The non smooth law between the input/output variables, for instance :

- Complementarity condition
- Relay
- Friction, ...

2.5   Formalization into a general abstract class

...

2.6   Description of the numerical strategy of resolution

In this section, the author describes the strategy of resolution. The goal of this section is not to describe precisely the basic algorithm we use, but how the general system is discretized and reformulated into a relevant form for numerical purpose. The goal of this description is to identify majors steps in the numerical resolution.

It will be interesting to provide a pseudo-algorithm of the numerical resolution or a flow-chart.

If a xxxlab simulation file exists, it must be added. This part must be closely related to the algorithm implemented in the xxxlab file.

2.7   Description of the relevant output

In this part, the author describes the relevant results for post-processing.

3   Guidelines for the xxxlab simulation file

...